Keyword extraction with KeyBERT: exploring and extending the approach

Barbara Frittella
Politenico di Torino
s332107@studenti.polito.it

Sara Mola
Politecnico di Torino
s331400@studenti.polito.it

Guido Spina
Politecnico di Torino
s319848@studenti.polito.it

***Abstract*—This project has the purpose to explore and extend the usage of KeyBERT, a method for the extraction of keywords and keyphrases from textual documents. KeyBERT works based on the semantic similarity between the candidate keywords and the text by leveraging BERT (or BERT-like language models), a bi-directional transformer model able to capture the semantical meaning of words.**

As a base language model we employed all-MiniLM-L6-v2 , the default model for KeyBERT. It is a fast, lightweight model efficient in the semantic search task.

For the first extension in our project we carried out a comparative assessment between three models: the base all-MiniLM-L6-v2 model, our fine-tuned version of all-MiniLM-L6-v2 and SciBERT , a BERT-like model trained on scientific papers similar to the ones we performed our evaluation on. The fine-tuning of the base model slightly improves the performances, whereas SciBERT performs worse than the other two models.

The second extension aims to improve the keyphrases selection of KeyBERT by removing those that do not make sense in human language (e.g. "machine learning with"). To do so, we apply Part-of-Speech tagging to the keyphrases and only keep those with an acceptable POS tagging. Although the performance on the evaluated metric did not improve significantly, human review of the keyword shows that they more closely resemble human annotated keywords.

The code and models used to carry out the project can be found in the repository https://github.com/s319848/DNLP-project-2025/

***Index Terms*—KeyBERT, keywords, extraction, embeddings, POS, keyphrases**

I. Problem statement

KeyBERT [1] is a minimal and easy-to-use method to extract keywords. It leverages BERT [2] embeddings to extract keywords (or keyphrases with a predefined n-gram length) based on the similarity with the embedding of the overall document, ranking them through three possible approaches: cosine similarity, Max Sum Similarity or Maximum Marginal Relevance. The key process is the generation of word and document embeddings, for which KeyBERT uses the `all-miniLM-L6-v2` [1] language model in its default version. We chose to evaluate the extraction of the keywords on two datasets: *Nguyen2007* and *SemEval2010*, taken from a collection of datasets with annotated keywords [2]. Both datasets contain papers related to computer science.

For our first extension, we wanted to explore the difference in results if we changed the language model used for the embeddings. So we compared the results obtained with the base model with a version of `all-miniLM-L6-v2` fine-tuned on a portion of the *Krapivin2009* dataset containing computed science papers, to see if we could adapt the model to a scientific domain. Additionally, we compare the results of using SciBERT [3] as a language model for embedding: since SciBERT is a model pre-trained on scientific corpus, we wanted to study if it was able to perform better than a model trained on a generic corpus.

The second extension arises from the manual observation of the keyphrases extracted by the KeyBERT base model in a preliminary study of the problem. We noticed that many keyphrases, although semantically related to the topic of the paper, were very different from what a human would produce as a keyphrase, meaning they often had a nonsensical structure. To solve this problem, we aimed at removing these nonsensical phrases by looking at which POS sequences are more frequent in the ground truth keyphrases of *Krapivin2009*. Then, only the extracted keyphrases with an acceptable POS tagging are retained. While this method may not improve significantly the similarity score evaluations, it might help to reduce human intervention in the annotation of a document with keywords and keyphrases, avoiding implausible annotations.

II. Methodology

A. Hyperparameter tuning

KeyBERT has two main parameters to choose: the length of the n-grams (minimum and maximum length) to consider for the keyphrases and the method used to evaluate the semantic similarity, which can be either cosine similarity, Max Sum or Maximum Marginal Relevance (MMR). We performed a grid search evaluation of all three similarity methods by considering both (1,2) and (1,3) for n-grams, performing an evaluation on the *Nguyen2007* dataset of the metrics in Tab. I. We evaluated precision, recall, F1, BERTScore F1, ROUGE-1, ROUGE-2 and ROUGE-L F1 by extracting 10 keyphrases and comparing them to the annotated keywords and keyphrases of the datasets.

The BERTScore library requires that the predictions and the references have the same number of keywords: therefore if the reference has more keywords than the prediction, the latter is padded by adding instances of the most relevant keyword. If instead the number of extracted keywords is higher than the

[1] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
[2] https://github.com/LIAAD/KeywordExtractor-Datasets

number of references, only the most relevant keywords are kept for evaluation.

The evaluation method remains the same for each method and approach in this project. To increase the significance of the evaluation, for the first and second extension we also measured performances on the *SemEval2010* dataset.

The best distance method was proved to be cosine similarity, with the scores obtained for both n-grams options being similar. We chose to select (1,3) as the length of the n-grams because it also includes n-grams of length 2, and because it allows for more flexibility for the second extension.

B. First extension

The first extension consists in the comparison between three language models used with KeyBERT for keyphrases extraction. The first model is `all-MiniLM-L6-v2`, the second one is a fine-tuned version of the first model, and the third one is SciBERT (in the uncased version of the model).

To test the base model we imported the KeyBERT library and selected `all-MiniLM-L6-v2` as a model, extracting the keyphrases for each document with the parameters found in the tuning phase.

Then, we fine-tuned `all-MiniLM-L6-v2` using a 20% subset of the domain-specific dataset *Krapivin2009*, which contains scientific papers annotated with keywords. The dataset was processed to create positive (keyword belonging to the document) and negative (keyword unrelated to the document) pairs. These pairs were used to train the model with a contrastive learning approach, optimizing a cosine similarity loss function. The training process was conducted for 3 epochs with a batch size of 8. The dataset was split into 80% for training and 20% for evaluation, with model performance monitored every 100 steps using an embedding similarity evaluator. A warmup phase of 100 steps was included to stabilize training.

SciBERT model was downloaded from the HuggingFace platform in its uncased version, and loaded inside KeyBERT to be used as a LM in the same way `all-MiniLM-L6-v2` was used.

C. Second extension

For the second extension we leverage Part-Of-Speech (POS) tagging to reject the keywords with an implausible structure in order to improve the linguistic quality and readability of KeyBERT extraction.

KeyBERT, using its base model, is prompted to extract, for each document in a given dataset, 30 (1-3)-grams keywords which are then associated with their corresponding Part-Of-Speech. The keyword's POS are compared to a list of acceptable ones (ranked according to KeyBERT's relevance score) and the first 10 matches comprise the definitive output of the extraction pipeline. If less than 10 keywords are retained, the final set is filled with the highly ranked discarded ones, i.e. those that were excluded due to their POS but were ranked highly by KeyBERT's relevance scoring.

Since POS tagging relies on contextual information that is not available for the extracted keywords, we applied Part-of-Speech analysis to the complete document, identified all the occurrences of a candidate keyword and assigned the most frequently occurring POS sequence as their definitive one. Keywords often appear in variations of their exact form, so we decomposed each candidate in unigrams and, for each unigram, identified the instances where the context contained lemmatized matches of the remaining terms in the original n-gram. The context is represented by a window with a standard length of 5 words that is increased for longer keywords.

For a quicker but less accurate POS tagging of the keywords we also provided a unigram-based method. This approach assigns to each word its most frequent POS tag from its occurrences within the document, disregarding the surrounding n-gram context.

The list of acceptable POS sequences is established using both methods described above by extracting a list of the 30 most frequent grammatical structures in the ground truth keywords of *Krapivin2009*, which was not used in the evaluation, to avoid generating bias.

III. Experiments

A. First extension

The fine-tuning of the model has been carried out with PyTorch, using the libraries provided by SentenceTransformers, with the method explained in the Methodology section. The total duration of the fine-tuning was 1149s, using a random sampling of 20% of the *Krapivin2009* dataset as data.

After the fine-tuning, the model has been manually saved and uploaded on the GitHub directory to be downloaded and evaluated on the *Nguyen2007* and *SemEval2010* datasets. The total duration of the evaluation can be seen in Tab.II.

SciBERT has been downloaded from HuggingFace with the `allenai/scibert_scivocab_uncased` release and loaded inside the KeyBERT method of the `keybert` library for use. It has then been evaluated on the same datasets as the other models.

The results of the evaluation of the three models can be seen in Tab. II.

Compared to the base model, the fine-tuned model outperforms it in several metrics (e.g. recall and F1 score), and performs equally in the remaining ones (e.g. BERTScore F1 or ROUGE-L F1) in both datasets. It never performs worse, although it takes longer to carry out the evaluation. This was the expected behavior: since the fine-tuning was arguably lightweight in terms of training time and documents, we did not expect to improve the baseline score significantly. Nonetheless, the improvement of several scoring metrics is proof that the fine-tuning has been performed correctly, and that a heavier fine-tuning could possibly improve the performance even more.

The performance of SciBERT was worse or equal than the baseline in all the metrics, with the exception of the F1 score in the *SemEval2010* dataset, which is slightly better.

TABLE I
VALUES OBTAINED FOR THE TUNING OF THE HYPERPARAMETERS

Distance method	N-grams	Precision	Recall	F1	BERT Score - F1	ROUGE Score - F1			Extraction time
						ROUGE-1	ROUGE-2	ROUGE-L	
Cosine similarity	**(1, 2)**	**0.06**	**0.07**	**0.06**	**0.87**	**0.42**	0.08	0.32	**100**
	(1, 3)	0.03	0.03	0.02	0.86	0.40	**0.09**	**0.33**	241
MaxSum	**(1, 2)**	0.04	0.04	0.04	0.86	0.36	0.05	0.26	1665
	(1, 3)	0.01	0.01	0.01	0.86	0.35	0.08	0.27	1860
MMR	**(1, 2)**	0.03	0.02	0.02	0.86	0.27	0.03	0.17	129
	(1, 3)	0.01	0.01	0.01	0.86	0.25	0.05	0.16	365

TABLE II
RESULTS FOR THE FIRST EXTENSION

	Nguyen2007								SemEval2010							
	Precision	Recall	F1	Bert Score F1	ROUGE Score - F1			Time	Precision	Recall	F1	Bert Score F1	ROUGE Score - F1			Time
					1	2	L						1	2	L	
Base model	**0.03**	0.03	0.02	**0.86**	**0.40**	0.09	**0.33**	**429**	0.02	0.01	0.01	**0.86**	**0.38**	**0.08**	**0.31**	**644**
Fine-Tuned	**0.03**	**0.04**	**0.03**	**0.86**	**0.40**	**0.13**	**0.33**	513	**0.03**	**0.02**	**0.02**	**0.86**	0.37	**0.08**	**0.31**	743
SciBERT	0.02	0.02	0.02	**0.86**	0.35	0.08	0.30	680	0.02	0.01	**0.02**	**0.86**	0.30	0.06	0.26	913

Despite the arguably small sample size for the evaluation, this result suggests that `all-MiniLM-L6-v2` is better suited for keyword and keyphrase extraction than SciBERT, despite the domain-specific training of the latter model and the bigger size (22M parameters vs 110M, the same as BERT), possibly due to differences in architecture, training procedure or training corpus size (3.17B tokens for SciBERT, whereas for `all-MiniLM-L6-v2` this is not available, we only know it has been fine-tuned on 1B sentence pairs[3]).

B. Second extension

The Part-Of-Speech analysis has been executed using the spaCy library. The extraction of candidate keywords is done though KeyBERT base model.

The accepted POS sequences list is determined by exploring a 20% subset of the *Krapivin2009* dataset, whereas the extension is evaluated on *Nguyen2007* and *SemEval2010*.

The POS tagging of all the documents requires 429s for *Nguyen2007* and 518s for *SemEval2010*. The duration of the keywords POS tagging by association to the text analysis is 1045s for the main function explained in the Methodology section, and 9s for the faster version. The results of the evaluation and the total required time are shown in Tab. III.

For both datasets, the main method slightly improves upon the base model in terms of precision, recall, and F1 score, while BERTScore and most ROUGE evaluations remain largely unchanged. The faster method provides minor improvements in recall and F1 (especially for *SemEval2010*). This result is compatible with our goal of an enhancement of the syntactic structure of the keyword, while having little impact on their semantic value.

Human assessment on a selected subset of the extracted keywords also validates an advancement in syntactic coherence and readability. For instance, both our methods successfully filtered out syntactically implausible or nonsensical keyphrases, such as

- *nano computing computational*
- *sorting compressed*
- *incremental mining sequential*
- *evaluators abstract*
- *photon mapping especially*

while *information retrieval exist* is an example of keyword that is discarded by the main method, but kept with the faster one.

Although both methods achieved equivalent results in our evaluation, one exhibited a significantly longer processing time. In practice, the choice between methods depends on the balance between precision and efficiency. Higher precision favors the main method, while efficiency favors the quicker one. Resource availability and dataset characteristics will ultimately determine the optimal choice.

IV. CONCLUSION

This project explored and expanded the capabilities of KeyBERT for keyword and keyphrase extraction from scientific documents, focusing on the impact of language models and improved syntactic coherence.

A comparative analysis was conducted using three distinct models: a base model (`all-MiniLM-L6-v2`), a fine-tuned model (`all-MiniLM-L6-v2` fine-tuned on a subset of the *Krapivin2009* dataset), and a domain-specific model (SciBERT). The results indicated that the fine-tuned model achieved marginal improvements in recall and F1 score relative to the base model, without any concomitant performance degradation. This suggests that even a limited degree of fine-tuning can enhance model performance within specific domains. Contrary to expectations, SciBERT did not outperform the base model and even underperformed in certain metrics, suggesting that model architecture and training procedures may be more influential than domain-specific pre-training for keyword extraction.

[3]https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

TABLE III
RESULTS FOR THE SECOND EXTENSION

	Nguyen2007								SemEval2010							
	Precision	Recall	F1	Bert Score F1	ROUGE Score F1			Time	Precision	Recall	F1	Bert Score F1	ROUGE Score F1			Time
					1	2	L						1	2	L	
Base model	0.03	0.03	0.02	0.86	0.40	0.09	0.33	**429**	**0.02**	0.01	0.01	**0.86**	0.38	0.08	0.31	**644**
Main method	**0.04**	**0.04**	**0.03**	**0.87**	**0.41**	**0.10**	**0.37**	1765	**0.02**	**0.02**	**0.02**	**0.86**	**0.39**	**0.09**	**0.32**	2581
Quick method	0.03	0.03	**0.03**	0.86	0.40	0.09	0.33	721	**0.02**	**0.02**	**0.02**	**0.86**	**0.39**	0.08	**0.32**	929

A Part-of-Speech (POS) tagging mechanism was implemented to improve the linguistic quality of the extracted keyphrases by filtering out syntactically implausible candidates. Acceptable sequences, based on an analysis of the POS patterns observed in ground truth keyphrases, were used to retain only extracted keyphrases with matching patters. This slightly improved precision, recall, and F1 score, while semantic (BERTScore) and syntactic (ROUGE) similarity scores remained largely unaffected. Human evaluation confirmed improved syntactic coherence and readability.

Despite the overall positive results achieved, it is important to acknowledge the limitations imposed by computational resources. Future research, unconstrained by these limitations, could explore more advanced fine-tuning techniques, leverage larger and more diverse datasets, and integrate additional linguistic features to provide deeper insights. Future research should also investigate the applicability of these approaches to other domains.

REFERENCES

[1] M. Grootendorst, "Keyword Extraction with BERT: A minimal method for extracting keywords and keyphrases," 2020. [Online]. Available: https://medium.com/towards-data-science/keyword-extraction-with-bert-724efca412ea.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805.

[3] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A Pretrained Language Model for Scientific Text," in *Proc. Conf. Empirical Methods in Natural Language Processing (EMNLP)*, 2019. [Online]. Available: https://www.aclweb.org/anthology/D19-1371.